Proceedings of THE 2nd INTERNATIONAL CONFERENCE ON

COMPUTATIONAL PHYSICS, Beijing, China, pp 522-525 (September 1993)

Lattice-Gas Automata on Parallel Architectures

Jeffrey Yepez
US Air Force, Phillips Laboratory, Hanscom Field, Massachusetts, USA
Guy P. Seeley*
Radex Corporation, Bedford, Massachusetts, USA
George Mou
Computer Science Department, Brandeis University, Waltham, Massachusetts, USA

September 12, 1993

Abstract

Conserved Navier-Stokes dynamics can be exactly simulated by lattice gas methods. This work studies several implementation issues of lattice gas automata on state-of-art parallel computer systems. We present performance results for the hexagonal LGA lattices on CAM-8 mesh, CM-5 fat-tree, and KSR1 hierarchical rings network topologies. The benchmarks range from 200 million to 500 million site updates per second.

I. Introduction

We have implemented lattice-gas automata (LGA) on three parallel architectures: MIT's CAM-8, Thinking Machine Corporations's CM-5, and Kendall Square Research's KSR1. For a 16-bit LGA our main findings are: (1) the CAM-8 appears to be much more efficient than the other two parallel architectures, delivering 25 million sites update per second per module; (2) the CM-5 can simulate larger lattices due its much larger memory sizes but can only deliver about 1 million sites update per second per node; and (3) the KSR1 can simulate about the same size lattice as the CM-5 and can deliver about 2 million sites update per node. Eight CAM-8 nodes, each clocked at 25 MHz with 16 custom memory management chips controlling 2^{22} 16-bit sites (8 Mbytes DRAM) using double buffered look-up table (2 Mbytes SRAM), are connected by a 3D mesh. 512 CM-5 nodes, each with a 32 MHz SPARC processor and four 16 MHz vector units with 32 Mbytes, are connected by a fat-tree network. 128 KSR1 nodes, each with a 64-bit 20 MHz processor with 32 Mbytes, are

*Supported by US Air Force Contract No. F19628-93-C-0023.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **12 SEP 1993**	2. REPORT TYPE	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Lattice-Gas Automata on Parallel Architectures**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Phillips Laboratory,Hanscom Field,MA,01731**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Conserved Navier-Stokes dynamics can be exactly simulated by lattice gas methods. This work studies several implementation issues of lattice gas automata on state-of-art parallel computer systems. We present performance results for the hexagonal LGA lattices on CAM-8 mesh, CM-5 fat-tree, and KSR1 hierarchical rings network topologies. The benchmarks range from 200 million to 500 million site updates per second.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**5**	

connected by a hierarchical ring. For the CM-5 and KSR1 machines the interprocessing communication cost is small relative to the internal streaming and collision computations for sufficiently large lattices. This suggests the relatively low communication bandwidth of the two commercial machines impose no serious performance degeneration. We also find a linear speed-up with increasing number of processors given large lattice sizes (e.g. 2k×1k lattice or larger).

LGA is a new approach to hydrodynamic fluid simulations[1]. It has found application to multiphase systems, reaction-diffusion systems, thermohydrodynamics, and flow through porous media. Particles, with mass m, propagate on a spacetime lattice with N spatial sites, unit cell size l, time unit τ, with speed $c = l/\tau$. A particle's state is completely specified at some time, t, by specifying its position on the lattice, $\mathbf{x}$, and its momentum, $\mathbf{p} = mc\hat{\mathbf{e}}_{\mathbf{a}}$ where lattice vector are $\hat{\mathbf{e}}_{\mathbf{a}}$ for $a = 1, 2, \ldots, B$. The particles obey Pauli exclusion since only one particle can occupy a single state at a time. The total number of configurations per site is 2^B. The total number of single particle states available in the system is BN. The lattice-gas cellular automaton equation of motion is $n'_a(\mathbf{x} + l\hat{\mathbf{e}}_{\mathbf{a}}, t + \tau) = n_a(\mathbf{x}, t) + \Omega_a(\vec{n}(\mathbf{x}, t))$, where the particle occupations and collisions are denoted by n_a andΩ_a, respectively.

The spatial coordinates of the lattice sites may be expressed as follows $\mathbf{x}_{ij} = \left(\frac{\sqrt{3}}{2}j, i - \frac{1}{2} \bmod {}_2 j\right)$ where i and j are rectilinear indices which specify the data memory array location used to store the lattice-gas site data. Given a particle at site (i, j), it may be shifted along vector $\vec{\mathbf{r}} = r\hat{\mathbf{e}}_{\mathbf{a}}$ to a remote site $(i', j')_a$ by the following mapping: $\left(i + \frac{r+1}{2} - \mathrm{mod}_2 j \bmod {}_2 r, j \mp r\right)_{1,4}$, $\left(i - \frac{r}{2} - \mathrm{mod}_2 j \bmod {}_2 r, j \mp r\right)_{2,5}$, $(i \mp r, j)_{3,6}$. All of our implementations use these streaming relations for address computations. In these streaming relations, the modulus operator is base 2 because a two-dimensional hexagonal lattice embedded into a square three-dimensional mesh is pleated.

II. Implementations on the CAM-8, CM-5, and KSR1

The desktop CAM-8[1] produced by Margolus and Toffoli of MIT[2] directly implements in hardware lattice-gas data streaming and collisions. The node-to-node communication architecture is a cartesian three-dimensional mesh. Each site of the lattice has a "pile" of bits (a multiple of 16). Each bit of the pile, or each bit plane, is "kicked" through the lattice in any directions. The specification of the x,y, and z components of the kicks for each bit plane exactly defines the lattice. The kicks determine all the global permutations of the data. Local permutations of data occurs within the bit piles are computed in double-buffered SRAM look-up tables (LUTs) that store all possible physical events with a certain input and output configurations. The width of the CAM-8 LUTs are 16-bits. This is a reasonable width satisfying the opposing constraints of model complexity versus memory size limitations for the SRAM. Site permutations of data wider than 16-bits must be implemented in several successive LUT

[1] The first 8-module CAM-8 prototype was operational in the fall of 1992.

passes. The CAM-8 has built-in 25-bit event counters for real-time measurements. We have used this feature to do real-time coarse-grain block averaging of the LGA number variables and to compute the components of the momentum vectors for each block. The data size of the coarse-grained data is sufficiently small that it can be transferred back to the front-end host for graphical display as an evolving flow field within an X-window.

In CAMForth, we implemented a two-speed hexagonal LGA with a rest particle, including gravitational forcing, free-slip and no-slip boundaries which may be oriented horizontally, vertically, or inclined ±60°, and heating and cooling sites to model temperature controlled boundary surfaces. This has been encoded within a single LUT. The ability of encoding such complex dynamics within 16-bits is one of the remarkable aspects of the LGA formalism in terms of efficient memory use affording us the ability to do flash updating from prestored collision tables. 98% of the 2^{16} entries in the collision tables are used in this model.

The Thinking Machines Corporations's CM-5 is a parallel computer that can contain up to 16384 processing nodes.[2] These processing nodes are all connected via a "fat-tree" communications net providing general inter-node communication. A front-end host, a modified SUN workstation, controls these processing nodes. A SPARC processor on each node issues instructions to the vector units and performs most bookkeeping tasks while the vector units perform arithmetic and logical operations on the data. Each vector unit has a peak rate of 32 million 64-bit ops (floating point or integer) for a combined total of 128 Mops/node. Each node's memory is divided into 8 Mbyte banks, one for each vector unit. Each vector unit has its own independent 128 Mbyte/sec path to memory for a combined memory bandwidth of 512 Mbyte/sec for each node. The DoD CM-5 at the Army High Performance Computing Research Center in Minneapolis, Minnesota contains 544 nodes for a total of 16 Gbytes of memory and 64 Gops of peak processing speed.

We have implemented an LGA simulator on the CM-5 using the high level parallel C*-language and the low-level C/CMMD/DPEAC language. In C*, the geometry of the problem is specified at the onset by defining the data structure's *shape*. This is usually a D-dimensional array with a certain size in each dimension. The shape definition defines all the needed communication topology for the compiler. It is possible to declare Boolean shapes in C*. The partitioning of the space between processors is handled completely by the C* compiler. Given a certain lattice size, for example 1024×2048, with have found the performance of the CM-5 to vary linearly with the number of processing nodes. This linear variation is expected so long as the lattice size is sufficiently large. We did several runs for lattice sizes $64 \times 128, 128 \times 256, \ldots, 8192 \times 16384$. For small lattice sizes, the performance is very poor, on the order of a million site updates per seconds. This is because processor to processor communication bandwidth limits the streaming rate. As the lattice size increases, the number of sites interior

[2] Currently the largest CM-5 resides at Los Alamos National Laboratory with 1024 nodes. Our work has been done on DoD's 544-node CM-5.

to the node grows and the number of sites on the partition boundary decreases. Consequently, the site update rate continuously improves with larger lattices. The update rate asymptotically approaches about 25 million site updates per second for a 256-node partition. This is equivalent to approximately 100,000 site updates per processing node. The C* compiler does not appear to be very efficient for the LGA algorithm.

In our C/CMMD/DPEAC implementation we partition the lattice into equally sized rectangles with one node (four vector units) assigned per rectangle. The streaming phase of the computation uses the address of each bit's destination in a pre-computed table. This is done so potentially complex addressing calculations are performed only once during initialization. Before a site can be updated, a communication phase must take place so each site can access all its neighbors. Communication must take place across node and vector unit boundaries. The communication is done so every site has access to all its neighbor values on one vector unit's 8 Mbyte bank of memory. After the streaming operation is complete the new site value is run through a LUT. Each vector unit has its own copy of the LUT, so this part of the calculation is not time consuming, yet it consumes are large amount of memory. We have achieved update rates of about 106 Msites/sec on a 128-node partition of the CM-5 for the FHP gas model. We find that the longer the system is across each node the greater the realized performance. This is due to the long system size across each node increases the fraction of sites in the interior of each vector unit and these sites do not need to communicate with sites on adjacent vector units or processing nodes.

The KSR1 is a parallel computer based on a shared-memory model. Sets of 32 processors are connected by a ring network with a communication bandwidth of 1GByte/sec between each set. Sets of rings of 32 processors are in turn connected in a ring at the next level. The processors are clocked at 20MHz. The peak performance per processor is 40 Mflops. Referencing data not physically located in one processor causes a *cache miss* and consequently brings in a cache line of 128 bytes. The cost of the cache miss depends on whether the data is located in the same ring or not and can cost up to several hundred machine cycles. Thus, it is crucial to exploit the locality of data in programming the KSR1.

We implemented an LGA simulator in KSR Fortran - an extension of Fortran with parallelizing KSR directives. The hexagonal lattice is embedded is a similar fashion to our implementation on the CAM-8 and CM-5. A $m \times n$ lattice is partitioned into p stripes of shape n by m/p, each assigned to one processor. The boundaries of the lattice are programmed separately and all the processors share the work. We have benchmarked the performance in two ways: varying the number of processors with fixed lattice size and varying the lattice size for fixed number of processors. The results show that a) the speedup is close to linear as we increase the number of processors and b) the per node site update rate increases with the lattice size but slowly decreases as the number of processor increases. The KSR1 can deliver about 2 million sites update per second per node for large lattices.

III. Conclusion

The CAM-8 architecture is an elegant, arguably the best, distillation of lattice gas dynamics that has been realized in low-cost desktop hardware. We look forward to the construction of a larger CAM-8, with much more than eight modules, in the near future. The CM-5 results are based on a beta version of the software and are not necessarily representative of the full version of the software.

JY would like to acknowledge Dr. Norman Margolus, MIT, for his extensive collaboration on the CAMForth LGA implementation.

Research time on the Connection Machine has been supported by, or in part by the Army Research Office, contract number DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center.

References

[1] G.D. Doolen, "Lattice Gas Methods: Theory, Applications, and Hardware", MIT/North Holland 1991

[2] N. Margolus and T. Toffoli, *Cellular Automata Machines* in the collection "Lattice Gas Methods for Partial Differential Equations", SFI SISOC, Eds. Doolen et al., Addison-Wesley Publishing Co., pg 219-249, 1990